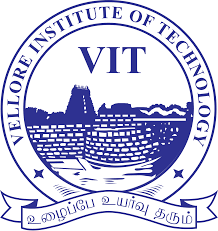
**Obtaining data from multifunctional meter using RS485 serial communication and MODBUS protocol**

Submitted By: **Matheesh M**

Registration No: **20BEE0021**



**School of Electrical Engineering**

**Embedded System**

**Design (EEE4020)**

**Final Review**

April 15, 2023

**Obtaining data from multifunctional meter using RS485 serial communication and MODBUS protocol**

Gautam Karthik V (20BEE0085), Matheesh M(20BEE0021),B K Sai Vijay(20BEI0052), Mourishwar I (20BEE0097), Sidrameshwar (20BEE0289) and Abhishek Nadagouda (20BEE0356)

School of Electrical Engineering, Vellore Institute of Technology, Vellore-632 014

***Abstract* —** Data communication based on Modbus protocol has seen its application in many areas such as healthcare, transportation, health automation etc. In this paper, it is explained how to obtain meter readings from a multi-functional meter using RS-485 serial communication. Here STM32 is used as master device, which obtains data from the slave device i.e. multifunctional meter in accordance with modbus protocol. Based on the given function code, various parameters such as voltage, current and frequency can be obtained.

***Index Terms* — Embedded, Modbus, RS485, Multifunction meter.**

1. INTRODUCTION

Modbus is a communication protocol widely used in industrial automation and control systems. Modbus uses a client-server model, where a master device (the client) requests data or performs actions from one or more slave devices (the server). The master device sends requests to the slave devices using a standard message format, which includes the slave address, function code, data, and error checking information. However, it should never be confused with a medium of communication. MODBUS only forms the messaging structure but is not the physical medium of data transmission.

RS-485 is a standard for serial communication in telecommunications systems. It is a physical medium of data transmission that not only supports Modbus protocol but also other protocols like Profibus DP, OPTOmux, and DH-485 etc. It is a half-duplex communication protocol that enables reliable communication between multiple devices over long distances at high speeds. It follows asynchronous communication, in which there is a lag between the time a message is sent and the time the recipient receives and understands it. RS-485 uses a differential signaling system, where data is transmitted over two wires with opposite polarity to reduce electromagnetic interference and improve noise immunity.

A multi-functional meter is a device which can measure various electrical parameters such as voltage, current, frequency, power factor, real and reactive power, phase angle, energy, apparent power and so on. These are useful for diagnosing a variety of electrical issues, including those with household appliances, commercial electronic equipment, power supply, motor controllers, and wiring systems, among others. By continuously detecting and monitoring energy use, it also contributes to lower energy costs. Moreover, it stores a backup of additional integrated parameters. Most of the modern multi-meters have an RS485 port which has a positive and negative side. These ports can be connected to a PC or a microcontroller for serial communication of data. For this paper, multifunctional meters such as Selec MFM384, Selec MX-300 -1 were used to collect voltage, current and frequency parameters.

The program required for serial communication of data from multi-function meter is entered in STM32Cube IDE. The program consists of data required to read various parameters from multi-function meter such as Slave address, function code, address of register to be read, number of registers to be read and Cyclic Redundancy Check Function(CRC) are entered in the code. The entered values is sent to the meter using RS485 serial communication standards, whose ports can be found in the meter. On receiving the command, the meter will return the desired data in accordance to the Modbus protocol by sending data length and register value for the required parameter.

|  |  |  |  |
| --- | --- | --- | --- |
| Slave Address | Function Code | Data | CRC |
| 1 byte | 1 byte | 0 up to 252 byte(s) | 2 bytes  CRC Low|CRC High |

|  |  |  |  |
| --- | --- | --- | --- |
| Master Request | | Slave Reply | |
| Field Name | Example | Field Name | Example |
| Modbus slave address | 0x2F | Modbus slave address | 0x2F |
| Function code | 0x03 | Function code | 0x03 |
| Address of the register to read | 0x03 | Data length in bytes | 0x02 |
| Address of the register to read | 0xF7 | Register value (MSB) | 0x02 |
| Number of registers (MSB) | 0x00 | Register value (LSB) | 0x2B |
| Number of registers (LSB) | 0x01 | CRC (MSB) | 0x XX |
| CRC (MSB) | 0x XX | CRC (LSB) | 0x XX |
| CRC (LSB) | 0x XX | - | - |

Diagram

Description automatically generated

1. METHODOLOGY

Modbus protocol follows master-slave principle, where a single device known as master controls one or more devices known as slaves. Based on this, The methodology can be divided as configuration of Master unit or Modbus client, configuration of slave unit(s) or Modbus servers.

1. Configuration of Modbus Master unit

The modbus master unit is used to request data from one or more connected slave devices. Here, STM32F407 discovery board is used as Modbus device. It is a 32-bit microcontroller which supports various communication protocols like UART, CAN, SPI, I2C, Modbus. This microcontroller accepts dats in TTL (Transistor to Transistor Logic) form. To collect data from the multifunction meter, an RS485 to TTL converter is required to convert parameters from meter, which is transmitted in RS485 standards, into TTL form.

The RS485 to TTL converter has pins A and B which is connected with positive and negative ends of RS485 ports in the multifunction meter respectively. The transmit and receive pins are configured and connected with STM32F407 board accordingly. The 5V supply for the converter is taken from microcontroller. Pins DE and RE are short circuited to provide same input values.

The STM32Cube IDE is used to upload the code to the microcontroller. The board used is selected in the sofware and a C file is created. To implelment RS485 serial communication using Modbus protocol, we require several header files such as modbus function, modbus crc to be attached in the include forlder along with some common files like math program etc.

To setup communication between pins of STM32 and TTL converter, we use USART hardware. Here functionalities such as transmission, reception, read and write functions are defined and implemented by accessing several registers in STM32 microcontroller.

1. Configuration of Modbus Slave Unit

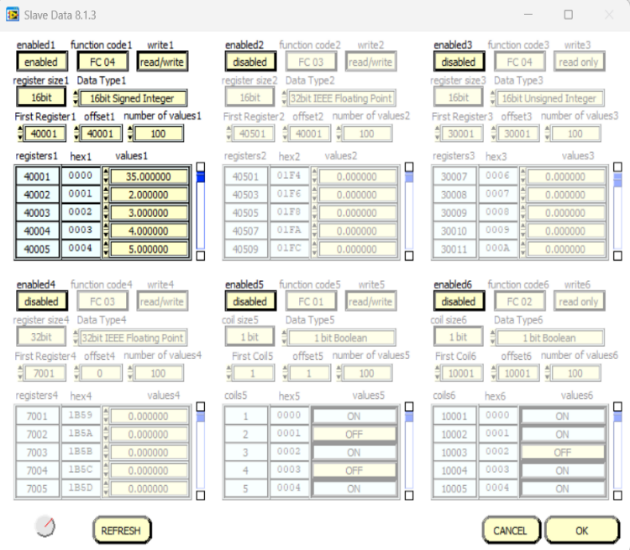
The modbus slave unit is used to send data to the master unit. Here Selec meter MX300-1-C-CE is used as the slave unit. It supports RS485 interface standard and MODBUS RTU protocol. It has communication address from 1 to 255. It supports half duplex transmission mode and float and integer data types.

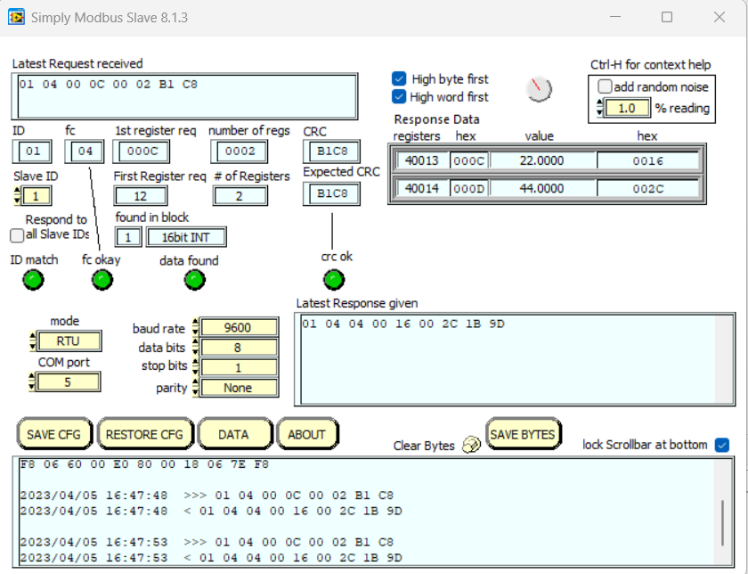
The maximum transmission distance is 500m. It supports transmission speeds from 300 to 19200 bps. It has None, Odd, Even parity and 1 or 2 stop bits. It has a maximum response time independent of baud rate of 500ms. It has an accuracy of 0.5% at full scale for both Voltage and Current measurements.

Selec meter MFM384 also supports RS485 interface standard and MODBUS RTU protocol. It has Device ID from 1 to 255 and None, Odd, Even parity. Transmission mode is half duplex and float and integer data types are supported. It has a maximum transmission distance of 500m. Transmission speeds can be varied from 1200 to 19200 bps. It has 1 or 2 stop bits and a maximum response time independent of baud rate of 100ms.

1. Testing Modbus Protocol using Simply Slave

To analyse the working of Modbus protocol using RS485 serial communication, we considered STM32 as Modbus Master Unit and Simply modbus software is taken as slave unit. Manual values are entered into the desired registers in that software. Proper sending and receiving of data in Modbus format ensures that Modbus communication works fine with the STM32 microcontroller.

****

****

1. OUTPUT RESULTS

**Code:**



Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

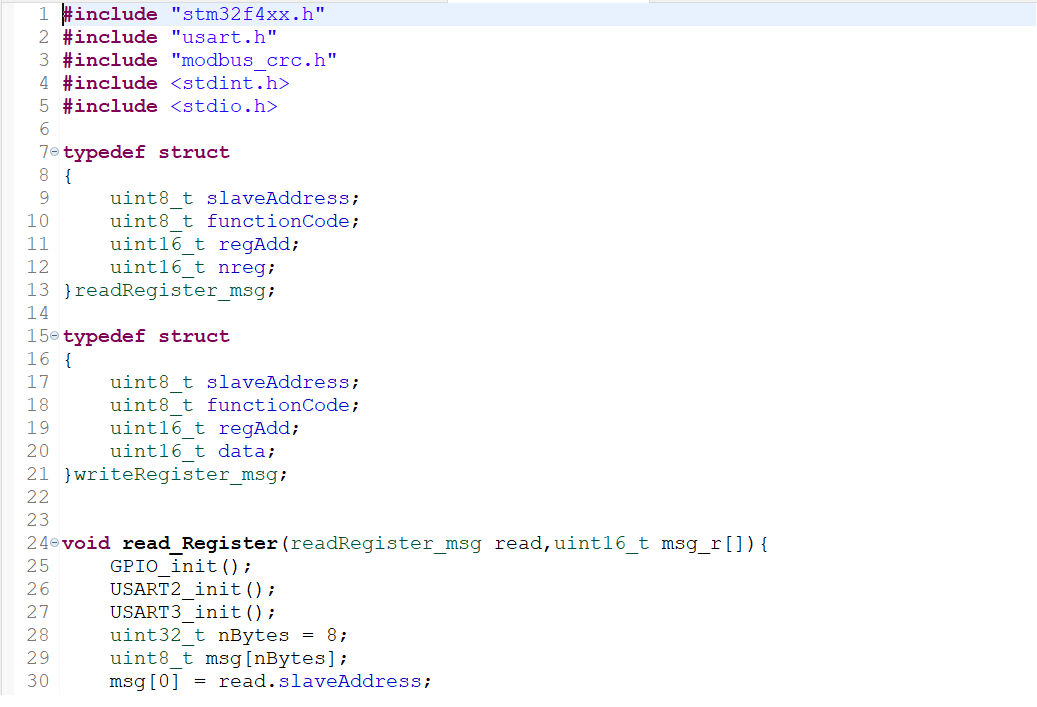
Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated



Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

A page of a book

Description automatically generated with medium confidence

After making the appropriate connections and running the code in IDE, the output value returned by the meter is in the Mid Little-Endian format. The required bits for obtaining parameters like voltage, frequency etc. are taken from this format message. The obtained bits which are in IEEE 754 form are then processed into a program to get the values as floating-point numbers. These numbers are then displayed continuously in the output window.

A picture containing text, electronics, computer, desk

Description automatically generated

Hardware Connections

1. CONCLUSION

Outputs values from the multifunction meter have been obtained according to Modbus Protocol, which is converted from IEEE 754 to floating point values for ease of usage. The desired parameters voltage and frequency are gathered using their respective function code values.

Since this data transmission is done using RS485 serial communication method, we can achieve higher data transmission speed, Low signal level interface, Good noise immunity in comparison with other serial communication protocols. Example: RS232.

The same methodology can be implemented in multiple meters using the same master setup to obtain electrical parameters. Each meter is differentiated by separate Slave IDs.

V. ACKNOWLEDGEMENT

We take this opportunity to thank our professor, Dr. K. Selvakumar for having faith on us and giving us an opportunity to put our hardware and theoretical skills to the test in a highly creative manner. He always provided valuable feedback and shared his knowledge with us so that we can effectively work on this project. We also thank him for providing us the components and for making necessary arrangements to complete this work.

VI. REFERENCES

[1] S., Basuli, K. and Sarma, S.S., 2008. Serial port data communication using MODBUS protocol. Ubiquity, 2008(January), pp.1-1.

[2] Thomas, G., 2008. Introduction to the modbus protocol. The Extension, 9(4), pp.1-4.

[3] Kuang, Y., 2014, September. Communication between PLC and Arduino based on Modbus protocol. In 2014 fourth international conference on instrumentation and measurement, computer, communication and control (pp. 370-373). IEEE.

[4] Pereira, R., Figueiredo, J., Melicio, R., Mendes, V.M.F., Martins, J. and Quadrado, J.C., 2015. Consumer energy management system with integration of smart meters. Energy Reports, 1, pp.22-29.

[5] Sandhya, C.S.R., Tirunagari, S.V., Sahoo, S. and Yemula, P.K., 2018, December. Extraction of Data from an RS-485 enabled Multi Function Meter for Building Monitoring Systems. In 2018 20th National Power Systems Conference (NPSC) (pp. 1-6). IE